

Adam Bańdo  
Władysław Marek Kolasa

## **Wstęp do baz danych**

### **(podstawy wiedzy dla studentów informacji naukowej i bibliotekoznawstwa)**

Zagadnienia:

- I. Wprowadzenie
- II. Co nazywamy bazą danych?
- III. Przydatna terminologia
- IV. Podział baz danych
- V. Modele a struktura baz danych
- VI. Projektowanie struktury bazy danych
  - A. Analiza teoretyczna (strukturalna i przepływu informacji)
  - B. Prace projektowe (definiowanie tabel i relacji)
  - C. Prace nad mechanizmami wyszukiwawczymi i interfejsem

### **I. Wprowadzenie**

Wobec wciąż postępującego rozwoju cywilizacyjnego i towarzyszącego mu postępowi technologicznemu nieustannie zwiększa się rola informacji. Dostęp do niej warunkuje nie tylko awans cywilizacyjny całych społeczeństw, ale decyduje o poziomie życia jednostek. W sferze edukacyjnej wpływa na poziom wykształcenia i otwiera możliwości samodoskonalenia. Tym samym otwiera perspektywy społecznego i zawodowego awansu, jednocześnie pozwala na realizację osobistych ambicji i zaspokojenie potrzeb, zarówno tych wyższych – intelektualnych, jak i zwykłych materialnych potrzeb życia codziennego. To ogromne znaczenie dostępu do informacji zostało już dawno zauważone i wykorzystane także komercyjnie przez wyspecjalizowane firmy. W sferze biznesu informacja stała się takim samym towarem jak inne wytwory ludzkiej działalności. Postęp technologiczny w końcu XX wieku pozwolił na sprawne gromadzenie informacji, jej opracowanie-przetwarzanie oraz udostępnianie. Rozwój komputeryzacji i globalnej sieci internetowej otworzył szeroki dostęp do źródeł informacji. Nastąpiła także rewolucja technologiczna jej form, możliwości przechowywania i sposobu udostępniania. Jest to szczególnie widoczne w przypadku baz danych, które przeszły ewolucję pod każdym z wymienionych względów. Papierowe bazy danych, kartoteki, ręcznie pisane fiszki oraz drukowane formularze ustąpiły miejsca ich

elektronicznym wersjom. Zmieniły się nie tylko forma i sposób utrwalania informacji. Najważniejszy z przełomów, wynikał z nowych możliwości technologicznego przetwarzania danych w informacje, zastosowania rozbudowanych narzędzi wyszukiwania oraz ich udostępniania na niespotykaną do tej pory skalę.

Przeciętny użytkownik informacji korzystający np. z Internetu często nie ma świadomości, że wynik jego poszukiwań jest rezultatem złożonego procesu przetwarzania danych. Nic w tym dziwnego, bowiem zainteresowany jest głównie rezultatem wyszukiwania. Nie zastanawia się nad tym, w jaki sposób informacja została wyodrębniona spośród innych zgromadzonych w bazie? Nie obchodzi go także, jaką pozycję zajmuje w jej strukturze? Jaka w ogóle jest struktura bazy, z której korzysta, czy jak wygląda przepływ informacji w bazie danych. Często też nie ma świadomości, że w ogóle korzysta z bazy danych. Wyjaśnieniu tych zagadnień służy niniejsze opracowanie.

## **II. Co nazywany bazą danych?**

Istnieje wiele definicji baz danych. Zagadnienie to wyjaśniane jest zarówno w szerokim jak i węższym znaczeniu. Najczęstsza odpowiedź udzielana przez studentów ogranicza się do stwierdzenia, że baza danych to „jakiś zbiór informacji”. Po zastanowieniu dodają, że jest to zbiór uporządkowany. Po części jest to prawidłowa odpowiedź, lecz niewystarczająca.

Pierwsza wątpliwość wyłania się ze stwierdzenia, że baza danych to zbiór informacji. Nasuwa się więc pytanie: gdzie? Na wejściu czy na wyjściu? Jaka jest różnica między informacją a danymi? Które z pojęć jest szersze i nadrzędne, a które węższe i podrzędne? I jeszcze jedno: skoro baza danych to zbiór informacji, to dlaczego nie używa się terminu bazy informacji w miejsce nazwy baza danych?

Niestety, dowolne „zonglowanie” pojęciami „informacja” i „dane” zdążyło się już utrwalić także w fachowych publikacjach na temat baz danych i właściwie stało się już normą. Zapomina się o fakcie, że zbiór danych tworzy informację. Pojęcie informacji ma więc charakter szerszy i nadrzędny. Prosty przykład: dane osobowe: imię i nazwisko, data urodzenia, adres zamieszkania, wykształcenie itp. tworzą informację o konkretnej osobie.

Cytowana wyżej definicja bazy danych nie uwzględnia jeszcze kilku ważnych zagadnień: rodzaju danych, sposobu ich porządkowania i przetwarzania oraz narzędzi wyszukiwania.

W tej sytuacji ograniczenie definicji bazy danych do stwierdzenia „zorganizowany zbiór informacji” jest odpowiedzią niepełną. Dlatego też konieczne jest jej rozwinięcie:

Baza danych (Data Base) to zbiór uporządkowanych i powiązanych ze sobą tematycznie danych utrwalonych na nośniku pamięci. Ponadto baza danych złożona jest z elementów o określonej strukturze np. rekordów. Aby definicja bazy danych oddawała jej współczesną (elektroniczną) postać, należy uzupełnić ją o pojęcie „mechanizmów zarządzania”, które określają zasady porządkowania, wyszukiwania, edycji i modyfikacji, czyli oprogramowanie do tworzenia i zarządzania bazami danych. Podsumowując: baza danych to nie tylko zbiór obiektów, ale także system zarządzania bazą danych (DBMS-Database Management System). DBAM to inaczej oprogramowanie użytkowe, które steruje-zarządza danymi w bazie. Zajmuje się m.in.: organizacją, przechowywaniem, odzyskiwaniem, ochroną i nadzorem nad integralnością danych. Może też formatować raporty, importować i eksportować dane oraz formułować zapytania (query) za pomocą własnego języka skryptów<sup>1</sup>.

### **III. Przydatna terminologia**

Mówiąc o bazach danych operujemy wieloma pojęciami, które mają duże znaczenie dla zrozumienia zagadnienia baz, ich modelu i struktury. Terminologia jest także istotna w procesie projektowania baz. Oto niezbędne pojęcia:

*Dane* – wartości przechowywane w bazie danych. Dane mają charakter statyczny w tym sensie, że zachowują swój stan do czasu ich zmodyfikowania ręcznie lub przez proces automatyczny.

*Informacje* – dane przetworzone, istotne dla użytkownika. W przeciwieństwie do danych, informacje mają charakter dynamiczny i wciąż ulegają zmianom w stosunku do danych przechowywanych w bazie. Prawidłowo skonstruowana baza to taka baza, która umożliwia odpowiednie przetworzenie danych w informacje użyteczne.

*Wartość zerowa (null)* – występuje wtedy, gdy dana wartość jest nieznana, lub po prostu jej nie ma.

*Tabela* – podstawowa struktura logiczna relacyjnej bazy danych. Każda tabela jest poświęcona konkretnemu tematowi, obiektowi czy zdarzeniu. Tabele składają się z rekordów (wiersze tabeli), rekordy z kolei dzielą się na pola pól (kolumny).

*Pole* – atrybut w teorii relacyjnych baz danych. Pola przechowują jednostkowe dane. Każde z nich reprezentuje pewną cechę tematu tabeli, w której występuje. Wyróżniamy pola segmentowe i wielowartościowe. *Pole segmentowe* zawiera więcej niż jeden typ wartości. *Pola wartościowe*, jak sama nazwa wskazuje, zawierają wiele wartości tego samego typu.

---

<sup>1</sup> P. Dyson. *Leksykon komputerowy*. Warszawa 1994.; A. Rydzewski. *Ilustrowany słownik techniki komputerowej*. Warszawa 1995.

*Rekord* – podstawowa jednostka informacji w bazie danych zawierająca pełny zestaw informacji o gromadzonych obiektach (np. opis bibliograficzny). Składa się z pełnego zestawu pól danej tabeli. Jest strukturą składową tabeli. Rekordy jednostopniowe dzielą się na pola. Rekordy wielostopniowe oprócz pól posiadają podpola.

*Relacje* – powiązania między parą tabel. (jeden do jednego, jeden do wielu, wiele do wielu).

*Perspektywa* – tabela wirtualna, czyli taka, która sama nie przechowuje danych. Pobiera je z tabel. Składa się z pól innych tabel.

*Kwerenda* – (ang. *query*) – zapytanie skierowane do bazy danych.

*Klucze* – pola pełniące specjalne zadania. Wyróżniamy *klucz podstawowy*, który będąc polem definiuje dany rekord w tabeli oraz *klucz obcy*, tworzący relacje między parą tabel.

*Raporty* - wydruki

*Makra* – inaczej skrypty, miniprogramy wspomagające wykonywanie czynności np. sterowanie formularzami.

*Formularze* – inaczej maski, interfejsy ekranowe, także formaty (sposoby prezentacji danych).

*Moduły* – pełnią tę samą funkcję co makra. Są stosowane do projektowania skomplikowanych operacji: obliczeń, skalowania, interfejsów. Inaczej fragmenty kodu źródłowego w języku programowania wysokiego poziomu.

#### **IV. Podział baz danych.**

Generalnie bazy danych dzielimy ze względu na sposób zarządzania. Stosując ww. kryterium wyodrębnić można *bazy operacyjne* i *bazy analityczne*.

*Bazy operacyjne* są wykorzystywane wszędzie tam, gdzie zachodzi potrzeba gromadzenia, przechowywania i co najważniejsze modyfikowania danych. Mówi się wtedy, że taka baza danych jest dynamiczna, ponieważ zawarte w niej informacje podlegają zmianom-uaktualnieniom. Taki typ bazy przechowuje informacje dynamicznie. Bazy operacyjne są szeroko rozpowszechnione i stosowane w codziennym funkcjonowaniu instytucji i firm, wszędzie tam, gdzie występuje stała zmienność gromadzonych danych.

*Bazy analityczne* charakteryzują się tym, że zgromadzone w nich dane mają charakter statyczny i nie ulegają zmianom.. Zawsze odzwierciedlają stan obiektów od pewnego ustalonego momentu. Dane nie są aktualizowane i nie odzwierciedlają stanu obecnego. Przykładem tego typu baz są bazy, które przechowują dane związane z pewnymi wydarzeniami, głównie o charakterze historycznym, retrospektywnym.

Biorąc pod uwagę rozmaite kryteria pragmatyczne, formalne i techniczne można też wyliczyć wiele innych podziałów baz danych (np. bazy uniwersalne / specjalistyczne, o dostępie plikowym / klient-serwer, bibliograficzne / faktograficzne itd.).

### **V. Modele a struktura baz danych.**

Mając na uwadze strukturę danych oraz sposób relacji między nimi wyróżniamy bazy o strukturze płaskiej (inaczej kartotekowe) i bazy relacyjne.

W modelu baz o płaskiej strukturze (np. popularny niegdyś ISIS i MAK) istnieje tylko jeden typ rekordów. Powtarzające się dane są fizycznie wpisywane do każdego rekordu – cechuje je zatem duża redundancja. Przykład bazy relacyjnej. Należy podkreślić, że bazy o tym modelu danych są trudne do reorganizacji i aktualizacji.

Rekord 1		Rekord 2	
<i>Pola</i>	<i>Wartości pól</i>	<i>Pola</i>	<i>Wartości pól</i>
V1. Autor	Nowak Jan	V1. Autor	Kowalska Maria
V2. Tytuł artykułu	Prasa gdańska w latach ...	V2. Tytuł artykułu	Prasa Wolnego Miasta...
V3. Tytuł czasopisma	Prace Bibliotekozn...	V3. Tytuł czasopisma	Prace Bibliotekozn...
V4. Miejsce wydania	Kraków	V4. Miejsce wydania	Kraków
V5. Rok wydania	1995	V5. Rok wydania	1996
V6. Numer czasopisma	5	V6. Numer czasopisma	9
V7. Strony	6-21	V7. Strony	12-23

Rys. 1. Przykład bazy płaskiej (kartotekowej)

W modelu baz relacyjnych (o strukturze relacyjnej) istnieje kilka źródeł danych (tabel) połączonych tzw. relacjami. Nie zachodzi „zjawisko” powtarzania się danych, gdyż te są zwykle wydzielone jako osobna tabela.

W nomenklaturze fachowej dotyczącej baz relacyjnych upowszechniło się określenie *relacyjny model logiczny bazy danych* (RMBD)

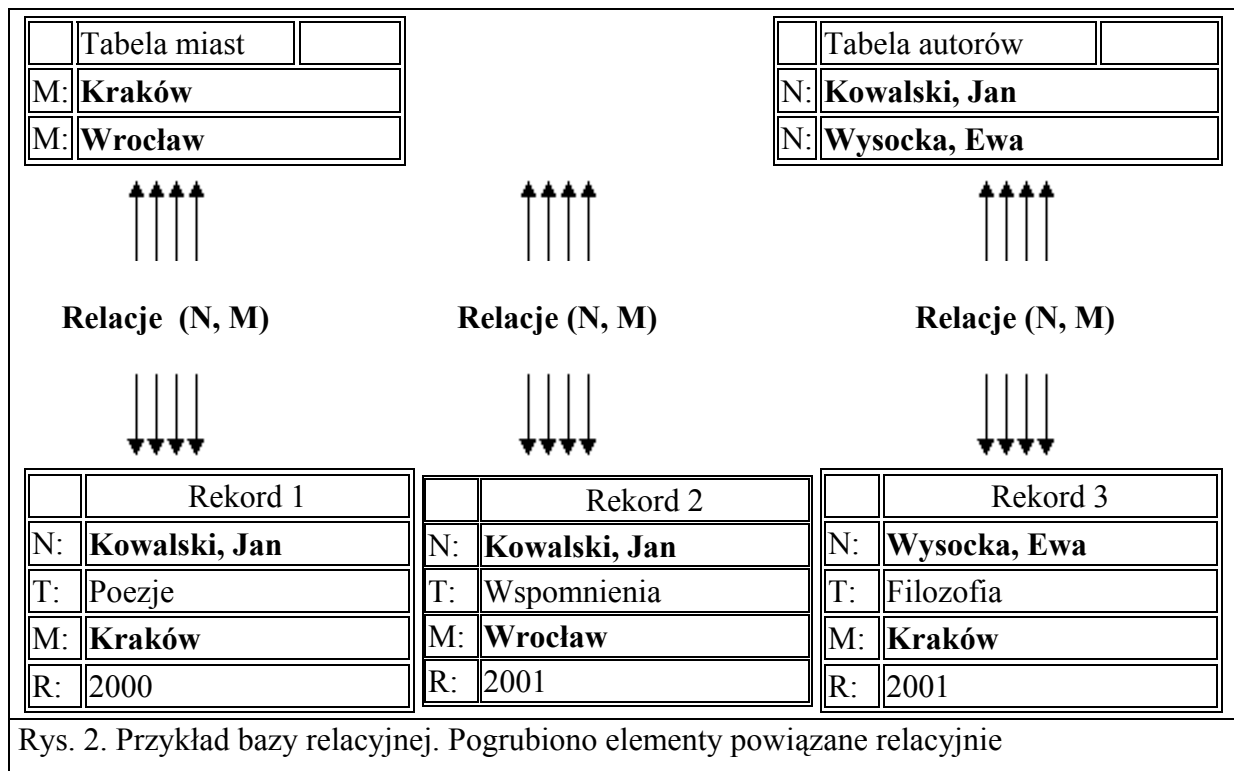
Zanim powstał współczesny relacyjny model logiczny bazy danych, do przechowywania i modyfikowania danych wykorzystywano hierarchiczne i sieciowe modele baz.

W hierarchicznym modelu bazy danych mają strukturę odwróconego drzewa. Aby uzyskać dostęp do danych, użytkownik zaczyna od korzenia, następnie nawiguje przez całe drzewo danych, które tworzą gałęzie, aż do interesującego go wyniku. Relacje w takim modelu okazały się jednak zbyt złożone i zaprzestano stosowania tego modelu w latach siedemdziesiątych.

Inny „historyczny” już model bazy to tzw. *sieciowy model logiczny*, który był udoskonalonym modelem hierarchicznym. Jego struktura także przypominała odwrócone drzewo, tyle tylko,

że istniała w tym przypadku możliwość występowania kilku drzew powiązanych ze sobą i dzielących gałęzie. Każde drzewo stanowiło część ogólnej struktury. Zaletą sieciowego systemu baz danych była możliwość tworzenia bardziej złożonych zapytań. Nadal jednak nie spełniał on wymagań użytkowników, którzy oczekiwali odpowiedzi na coraz bardziej złożone pytania. Umożliwił to dopiero relacyjny model logiczny bazy danych, oparty na dwóch gałęziach matematyki: na teorii mnogości i rachunku predykatów pierwszego rzędu.

W relacyjnym modelu logicznym (RMBD) dane przechowywane są w domenach, czyli tabelach. Każda tabela składa się z rekordów, czyli krotek. Rekordy z kolei składają się z pól. Kolejność rekordów pól w tabeli jest bez znaczenia. Każdy rekord wyróżniany jest przez jedno pole zawierające unikatową wartość. Zaletą tego modelu, w odróżnieniu od poprzednich, jest to, że użytkownik nie musi znać struktury bazy, fizycznego położenia rekordu.



Jeżeli potraktujemy bazę typu RMBD jako zbiór tabel, to pomiędzy tymi tabelami istnieją relacje, czyli powiązania, ponieważ dzielą ze sobą pola zawierające odpowiadające sobie wartości. W RMBD, jak wspomniano, relacje występują w trzech postaciach: jeden do jednego, jeden do wielu i wiele do wielu.

Dostęp do danych uzyskuje się podając nazwę interesującego nas pola oraz tabeli, do której to pole należy. W tym celu wykorzystuje się język SQL (Structured Query Language). SQL jest

standardowym językiem powszechnie używanym do wprowadzania, modyfikowania i odczytywania informacji z bazy danych.

Relacyjny model bazy danych cieszy się obecnie największą popularnością wśród innych modeli baz. Pod koniec XX wieku zarzucano mu, że oparte na nim aplikacje działają zbyt powolnie. Tak więc był to zarzut odnoszący się nie tyle do modelu bazy, ale do technologii. Współczesny poziom technologiczny pozwolił na ukazanie skuteczności zastosowania takiego modelu bazy.

Do zalet relacyjnego modelu baz należą m.in.:

Wielopoziomowa integralność danych (integralność znaczeniowa: poprawność, spójność oraz dokładność danych przechowywanych w bazie). Integralność na poziomie tabel gwarantuje, że pole identyfikujące każdy rekord w danej tabeli ma zawsze unikatową wartość i nigdy nie jest puste.

Z kolei integralność na poziomie pól zapewnia poprawność struktury każdego pola oraz logiczność zawartych w nim wartości. Ponadto wszystkie pola tego samego typu są zdefiniowane identycznie w całej bazie.

Integralność na poziomie relacji (integralność referencyjna) gwarantuje, że wszystkie relacje są poprawnie zdefiniowane, a dane w powiązanych tabelach są ze sobą zsynchronizowane.

Model relacyjny zapewnia także dokładność i poprawność danych dzięki wspomnianej wielopoziomowej integralności oraz łatwy dostęp do danych, które można odczytać z pojedynczej lub całej grupy powiązanych tabel.

## **VI. Projektowanie bazy danych**

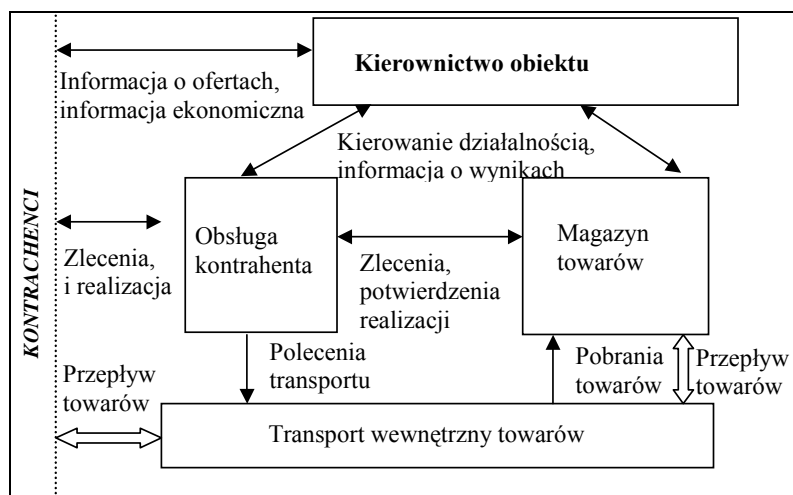
Projektowanie bazy danych jest czynnością, która wymaga systematycznego podejścia. Prace te można podzielić na trzy etapy: analizę teoretyczną (funkcje, elementy, przepływu informacji); następnie prace projektowe (definiowanie tabel, relacji) oraz prace nad mechanizmami wyszukiwawczymi i interfejsem.

### **A. Analiza teoretyczna (strukturalna i przepływu informacji)**

Pierwszym krokiem na tym etapie jest tworzenie jej projektu bazy. Warto na wstępie podkreślić, że baza danych, podobnie jak większość projektów komputerowych, jest modelem wycinka świata rzeczywistego, utworzonym tak, aby był możliwy do zapamiętania przez system informatyczny i zawierał optymalną ilość informacji do zastosowania, jakiemu będzie

służył. Aby osiągnąć te cele należy zastosować dowolną metodologię modelowania (np. strukturalną lub pragmatyczną). W praktyce najczęściej stosuje się podejście systemowe. W ujęciu takim – zgodnie z teorią systemów – zakłada się swoistą ontologię, w której kluczowe znaczenie mają cztery aspekty: 1. analizowany model składa się z elementów i specyficznych relacji między nimi oraz otoczenia systemu; 2. zakłada się samopodtrzymanie całości w zmieniających się warunkach; 3. zakłada się, że system przekształca się i tworzy nowe konfiguracje w odpowiedzi, na zmieniające się wymogi otoczenia; oraz 4. relacje pomiędzy elementami a otoczeniem mają charakter zwrotny. Schemat ów – jak łatwo zauważyć – tworzy wygodne podłoże do analizy obiegu i przepływu informacji.

Analizę należy rozpocząć, od zgromadzenia wiedzy na temat modelowanej rzeczywistości i wyobrazić ją jako schemat. Należy zatem zgromadzić informację o elementach oraz hierarchiach i powiązaniach między nimi. Na przykład przy projektowaniu bazy danych do obsługi przedsiębiorstwa na pierwszym etapie jest rozpoznawana struktura organizacyjna oraz obieg dokumentów; następnie rozpoznawane są informacje na wejściach i wyjściach oraz funkcje poszczególnych jednostek. Wygodnym narzędziem do dalszej analizy jest model „przepływu komunikatów”, czyli struktura, w której należy zdefiniować: 1. nadawców; 2. odbiorców; 3. zbiór komunikatów. Graficzny przykład takiej analizy zawiera poniższy rysunek (rys. 3).

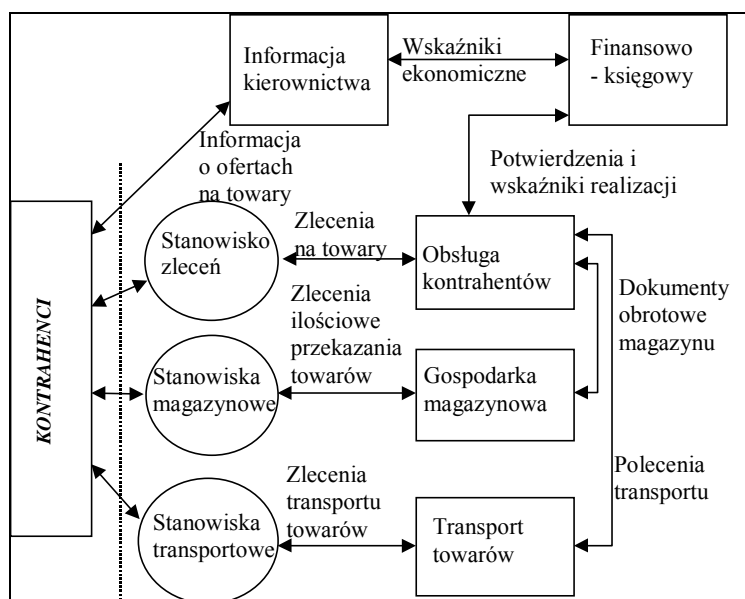


Rys. 3, Struktura przykładowego obiektu – źródło: <http://www.woiz.polsl.pl/~senczyna/>

Na dalszym etapie projektowania obiekt należy poddać konkretyzacji, tak aby uzyskać powiązania informacyjne między elementami obiektu i otoczeniem oraz wyspecyfikować realizowane cele i zadania. Czynności te należy wykonać dla każdego: 1. elementu systemu



(np. stanowiska pracy); 2. transakcji; 3. zadania cząstkowego oraz powiązać je w jeden spójny diagram. Postępowanie to ilustruje rysunek 4.



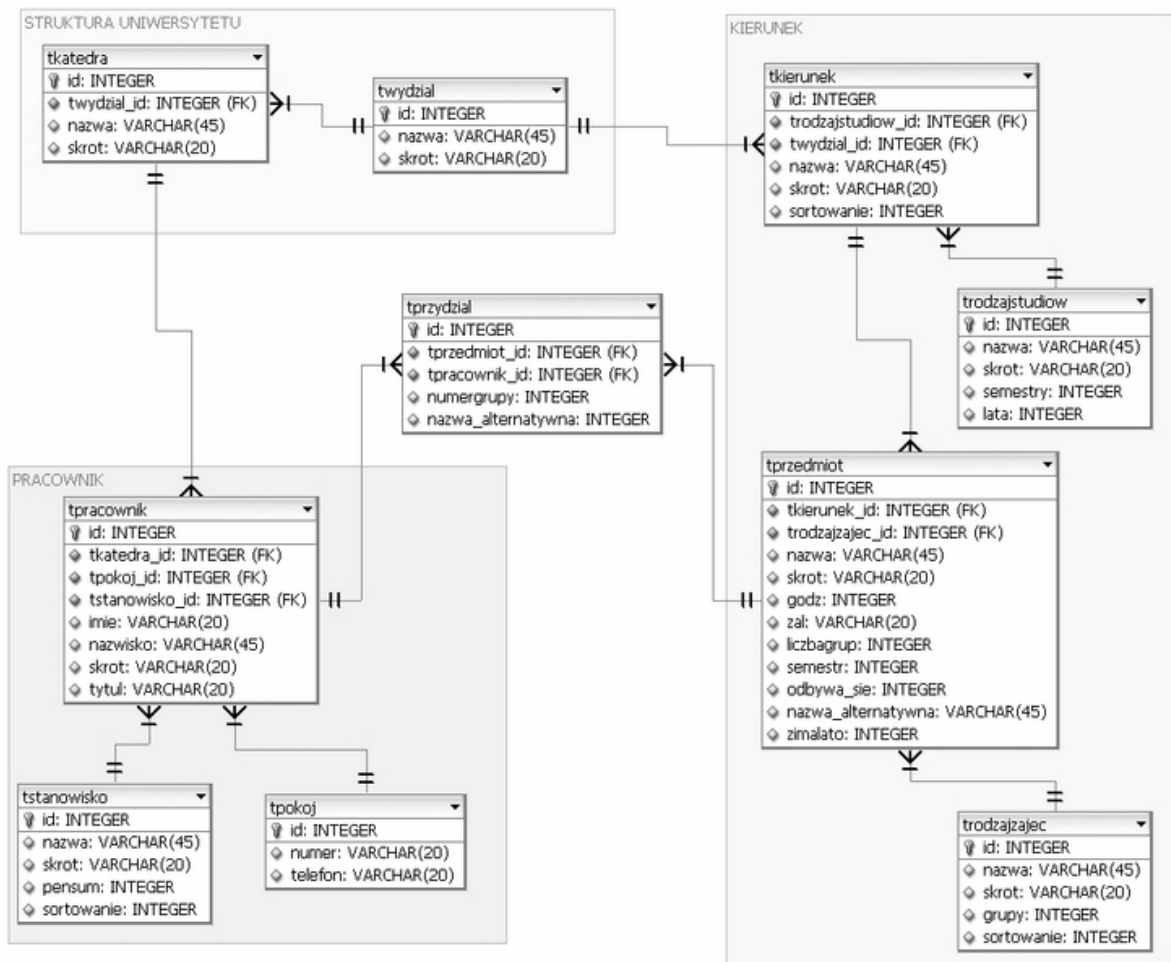
Rys. 4, Model obiektu – diagram szczegółowy – źródło: <http://www.woiz.polsl.pl/~senczyna/>

## B. Prace projektowe (definiowanie tabel i relacji)

Uzyskane opisaną wyżej drogą informacje stanowią punkt wyjścia do skonstruowania zbiorów danych skojarzonych z przepływami informacji, czyli modułowego modelu bazy danych. Model ten zawiera konkretne typy danych i transakcji (oraz wejść i wyjść), które pozwalają na zaprojektowanie elementarnych jednostek bazy danych, czyli tabel i określenia relacji między nimi. Tabele to podstawowe jednostki logiczne, które zawierają określone dane (np. informacje o dokumentach, podmiotach i przedmiotach transakcji oraz dane o transakcjach – np. daty); relacje zaś symbolizują sieć ich wzajemnych powiązań (najczęściej: obiekt-transakcja, obiekt-obiekt, obiekt-atrybut). Przykładami relacji są in. o klientach i dokonanych przez nich transakcjach określonym czasie i określonej ilości.

Wśród wielu zasad, które należy brać pod uwagę na etapie projektowania, kluczowe znaczenie ma dbałość o integralność danych i unikanie redundancji. Innymi słowy: w dobrze zaprojektowanej bazie jednorodne dane **nie powielają się** ani w odrębnych tabelach, ani w odrębnych rekordach (dążymy do minimum redundancji) oraz **są jednoznacznie identyfikowane** (najczęściej służy do tego celu specjalny numer identyfikacyjny); określone pola rekordu w zależności od zawartości są zdefiniowane jako **odpowiedni typ danych**

(tekst, liczba, waluta itp.); **relacje natomiast jednoznacznie określone zarówno pod względem kierunku i siły więzów**. Wyliczone elementy struktury bazy danych symbolizuje rys. 5. Tabelami są tu np. dane o pracownikach i kierunkach (wydziałach) uniwersytetu; relacjami np. przydział zajęć określonego pracownika z określonego przedmiotu dla określonej grupy studentów (w określonym miejscu, czasie itd.); typem relacji pomiędzy pracownikiem a przydziałem – relacja „jeden do wielu”, a zapośredniczona relacja przedmiotów do pracowników – to relacja „wiele do wielu”; typami danych są tu np. integer (cyfra długa) lub tekst o określonej maksymalnej długości (np. varchar [20]).

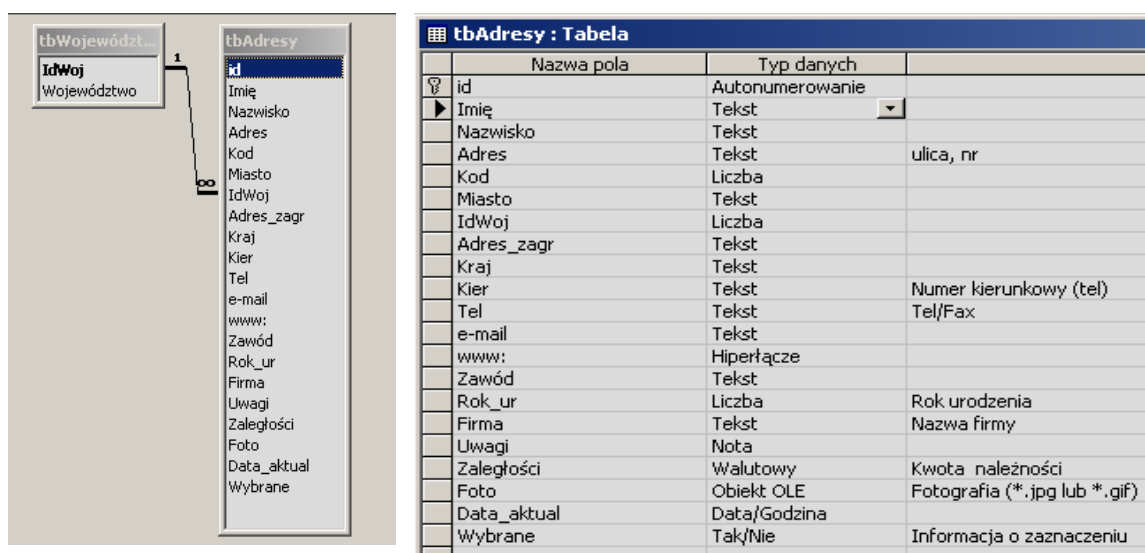


Rys. 5, Diagram bazy danych – źródło <http://serwis.magazynyinternetowe.pl>

Prostsze bazy danych mają zwykle definiowane mniej pól danych i relacji. Trywialnym przykładem może być osobista baza teleadresowa<sup>2</sup> (rys. 6), gdzie wyróżniono dwie tabele: tbAdresy (gromadzącą większość pól danych) oraz tbWojewództwo – zawierającą wykaz województw. Logika podziału wynika tu z analizy teoretycznie

<sup>2</sup> Baza wraz z komentarzami jest materiałem uzupełniającym do niniejszego artykułu. <http://ultra.ap.krakow.pl/~wmkolasa/Access/index.htm> [2008-10-12].

powtarzających się danych – zaś wykaz województw jest niewielkim zbiorem skończonym. W rozważnym przykładzie można by oczywiście wydzielić jako tabele inne pola (np. nazwy miejscowości lub zawody), ale winno być to uzasadnione ekonomią „częstości występowania” lub konieczności aktualizacji. Warto dodać, że skutkiem wydzielenia odrębnych tabel jest m.in. redukcja redundancji, co skutkuje też wzrostem integralności (szczególnie jeśli dane występują w wielu tabelach). Z drugiej zaś strony taka organizacja struktury (szczególnie wtedy, gdy zdefiniujemy silne więzy integralności) powoduje narzucenie sztywnych reguł postępowania na etapie tworzenia (dane wprowadzany w określonej kolejności), co skutkuje wzrostem pracochłonności w porównaniu z bazami o płaskiej strukturze danych.



Rys. 6, Prosta baza danych – relacje i widok projekt głównej tabeli

### C. Prace nad mechanizmami wyszukiwawczymi i interfejsem

Ostatnim etapem prac na bazą danych (oczywiście jeśli pominąć etap testowania i debugowania) jest zaprojektowanie odpowiednich mechanizmów wyszukiwawczych i interfejsu. Warto jednak podkreślić, że z informatycznego punktu widzenia są to elementy fakultatywne – bowiem z określoną bazą danych mogą współpracować różne interfejsy, może także baza nie mieć jawnego interfejsu i być częścią innego systemu, który komunikuje się z nią za pomocą określonego języka zapytań, np. SQL. W zależności od rodzaju bazy danych i potrzeb użytkowników mechanizmy te wyszukiwawcze mogą być rozmaite.

Do przeglądania zawartości bazy (ang. browse) służą tzw. indeksy (na stronach www często realizowany w formie hiperłączy). Mechanizm browse ma jedno poważne ograniczenie: pozwala zadawać zapytania tylko według jednego kryterium; z drugiej strony jego niepodważalną zaletą jest jawny wykaz terminów, niejako podpowiadanych przez system – które umożliwiają np. nawigację po strukturze drzewiastej. Część systemów, aby przyspieszyć proces przeglądania w dużych zbiorach tworzy pliki indeksowe w osobnych zbiorach uaktualnianych okresowo.

Do typowych zapytań (ang. search) służą mechanizmy zwane kwerendami. W tym przypadku użytkownik nie może liczyć na odpowiedź systemu i określony termin (frazę) musi podać samodzielnie. Z drugiej strony korzystanie z tej opcji (w odróżnieniu od browse) pozwala na stosowanie operatorów maskowania (położenia i in.) oraz łączyć zapytania operatorami Boole'a. Niektóre systemy są dodatkowo wyposażone w mechanizm rankingowania wyników zależnie o zdefiniowanego współczynnika trafień (podobnie jak w popularnych wyszukiwarkach). Innym możliwym udogodnieniem jest w tym trybie wyszukiwania systemowy tezaurus, który wykonuje odpowiednie przekierowania w czasie rzeczywistym (np. MeSH w Medline).

Kwerendy mogą też pełnić funkcje inne niż interfejs wyszukiwawczy: w niektórych systemach możliwe jest zdefiniowanie np. kwerend aktualizujących lub krzyżowych. Te ostatnie służą na łączenia prowadzenia rozmaitych obliczeń na danych (np. generowanie korelacji, zliczanie, a także tworzenie tabel do wykresów i in.); są one szczególnie użyteczne w analizie statystycznej dużych jednorodnych zbiorów (np. ankiet, kart wyborczych lub danych bibliograficznych).

Osobliwą odmianą kwerendy jest tzw. praca w trybie eksperta (ang. expert search). W tym przypadku wysoko wykwalifikowany użytkownik zadaje zapytania do bazy z konsoli. Rozwiązanie to stwarza ogromne możliwości wyszukiwawcze, ograniczone bodaj możliwościami samego języka kwerend. Z kolei jego wadą jest stosunkowo duży nakład pracy konieczny do opanowania leksyki i składni języka określonej bazy danych (najczęściej bowiem bazy udostępniają rozmaite języki dedykowane, np. Apache Lucene - Query Parser Syntax).

Komunikację z bazą danych umożliwiają interfejsy. Zależnie od potrzeb można zaprojektować jeden lub kilka interfejsów, albo – o czym wspomniano wyżej – nie

projektować żadnego. Popularną formą interfejsu dla większości baz danych jest dedykowany klient GUI lub interfejs webowy. W tym modelu klient pełni tu rolę pośrednika między użytkownikiem a bazą danych; umożliwia temu ostatniemu konwersację z bazą w języku naturalnym, sam zaś komunikuje się z bazą w języku niższego poziomu (np. SQL, z39.50, ODBC). W sensie praktycznym projektowanie interfejsu wiąże się najczęściej z analizą oczekiwań lub uprawnień użytkowników. Aby zaspokoić użytkowników pierwszego typu (użytkowników biernych) projektuje się ergonomiczne, elastyczne interfejsy umożliwiające daleko idącą personalizację. W interfejsach projektowanych z punktu widzenia uprawnień (dla użytkowników czynnych) akcent przesunięty jest z ergonomii na rygory bezpieczeństwa.

### **Bibliography:**

- R. Dobson: *Programowanie Microsoft Access 2000*. Warszawa, 2000
- P. Dyson. *Leksykon komputerowy*. Warszawa 1994
- M. J. Hernandez, *Bazy danych dla zwykłych śmiertelników*. Warszawa, 2004
- R. Riordan, *Projektowanie systemów relacyjnych baz danych*. Warszawa, 2000
- A. Simpson, E. Olson, *Access 97*. Gliwice, 1998